BLACK
CODHER

CODING PROGRAMME

CODING BLACK FEMALES

Black Codher Bootcamp

NÍ YỌ̀

BLACK CODHER

CODING PROGRAMME

UNIT 4 - Session 1
React

Goals for Unit 4 - Session 1

1. Understand what React is

2. Understand the difference between a websites, web applications and PWAs

3. Reference the React library in a Javascript application

What is React?

What is React?

Library not a Framework

- React is a JavaScript library for building fast and interactive interfaces
- Developed in 2013 by Facebook
- Other frameworks include Angular and Vue
- Most popular JavaScript library for building interfaces

What is React?

How Does it Work

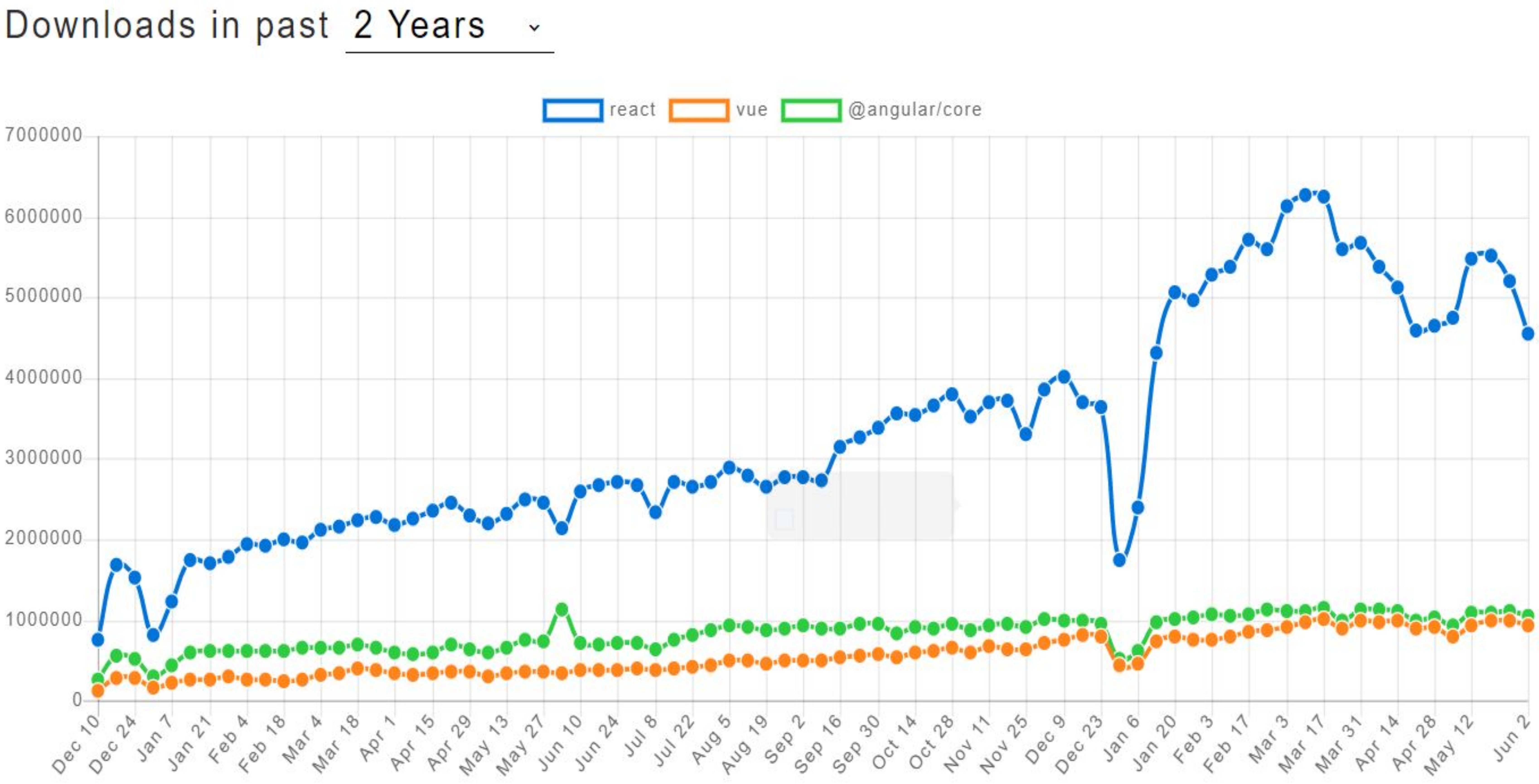
- With React you compose user interfaces using small isolated pieces of code called "components"

- React mainly responds to changes in the **Virtual DOM** by updating the component in the real **DOM**

- Unlike Angular (which is a JavaScript framework), React is relatively light-weight and requires other libraries in order to work

React Popularity?

- JavaScript Framework/Library Popularity (2018-2019)



Downloads in past 2 Years

Image credit: https://gist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190

React in Production

- Large companies that are using React in production:

- Netflix
- Facebook
- Instagram
- Airbnb
- Cloudflare
- Dropbox
- BBC
- .... (and many more)

Website, Application, PWA

Website vs Application vs PWA

What is a Website?

- Collection of pages on the world wide web that contains specific information
- The pages are comprised of codes that describes the layout, format and content on a page
- Websites are composed of **HTML**, **CSS** and sometime **JavaScript**
- Mostly consists of static content

Website vs Application vs PWA

What is a Web Application?

- A web application is designed for **interaction** with an **end user**
- There is a lot of overlap between a website and web application
- Web application often need authentication (the ability to log in) and have higher more **complex functionality** than websites

Website vs Application vs PWA

What is a Progressive Web Application (PWA)?

- Best of both worlds? A combination of native and web apps.
- The inner workings of a PWA are like web applications but installs like a native app and takes advantage of device functionality.
- At their core they are web applications that are based on standard web technologies.
- They utilise the newest JavaScript features making them feel and function like a native Android or iOS app.
- Disadvantage is that they are not that widely adopted*

React: What do I need to code?

- To code in React you will need an understanding of **HTML**, **CSS** and **JavaScript**.

- For this tutorial you will need to install **Node** and **Visual Studio Code**.

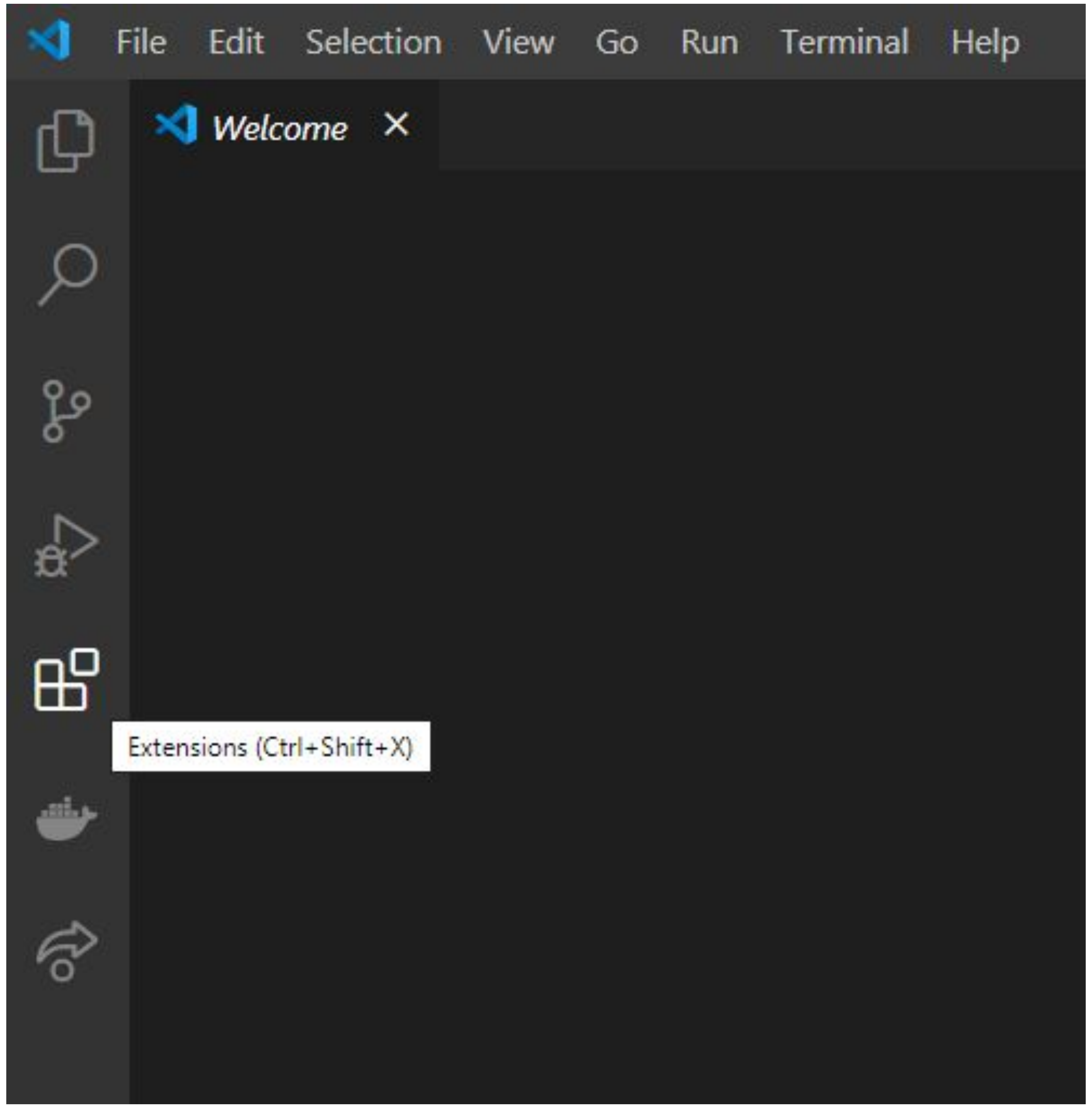
1. Node: https://nodejs.org/en/download/
2. Visual Studio Code: https://code.visualstudio.com/download

React: What do I need to code?

- For syntax highlighting and formatting Install the **Visual Code Extensions***:

  1. **Prettier** – Code Formatter (Esben Petersen)
  2. **Simple React Snippets** (Burke Holland)

- To install, open Visual Code:
- Navigate to Extensions (**Ctrl+Shift+X** in Windows)
- Alternative click the extensions icon ⊞
- Search for the extensions above and install.



Introduction to React

Thinking in React

1. To understand what React is it may be helpful to add React to an existing HTML page.

2. A very basic HTML page can consist of the following (it should look familiar):

3. The page consists of a head and body with a simple **<div>** with an id of "**root**".

```
<html>
    <head>
        <title>My Basic HTML Page</title>
    </head>
    <body>
        <div id="root"></div>
    </body>
</html>
```

Thinking in React

4. To add React to this page you can add the script files to the body, just before the closing **</body>** tag:

```
<script src="https://unpkg.com/react@16/umd/react.development.js" crossorigin></script>
<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js" crossorigin></script>
```

Thinking in React

5. This will give you the following page. This adds the basic library files to the page:

```
<html>
    <head>
         <title>My Basic React Page</title>
    </head>
    <body>
         <div id="root"></div>

     <!-- Load our React component. →
     <script src="https://unpkg.com/react@16/umd/react.development.js"...
     <script src="https://unpkg.com/react-dom@16/umd/react-dom.devel...
    </body>
</html>
```

Thinking in React

6. Next create a JavaScript file or add inline React script to start using the library.

   To create a script file, create a new file called **index.js** and reference before the **</body>** tag e.g.:

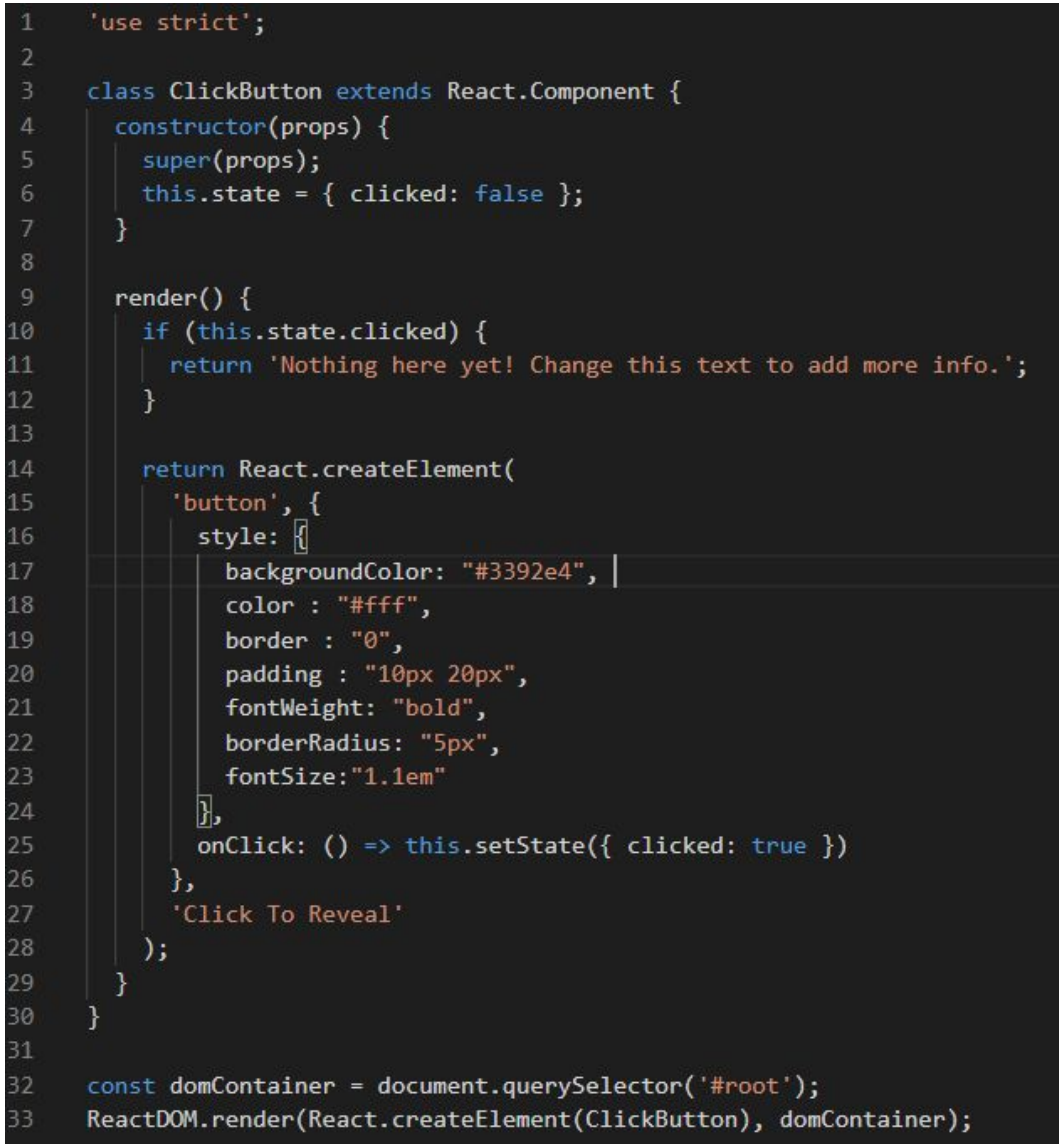
```
<script type="text/javascript" src="index.js"></script>
```

Thinking in React

- Add the following code to the **index.js** script:

- This is an example of a **React.Component** class.

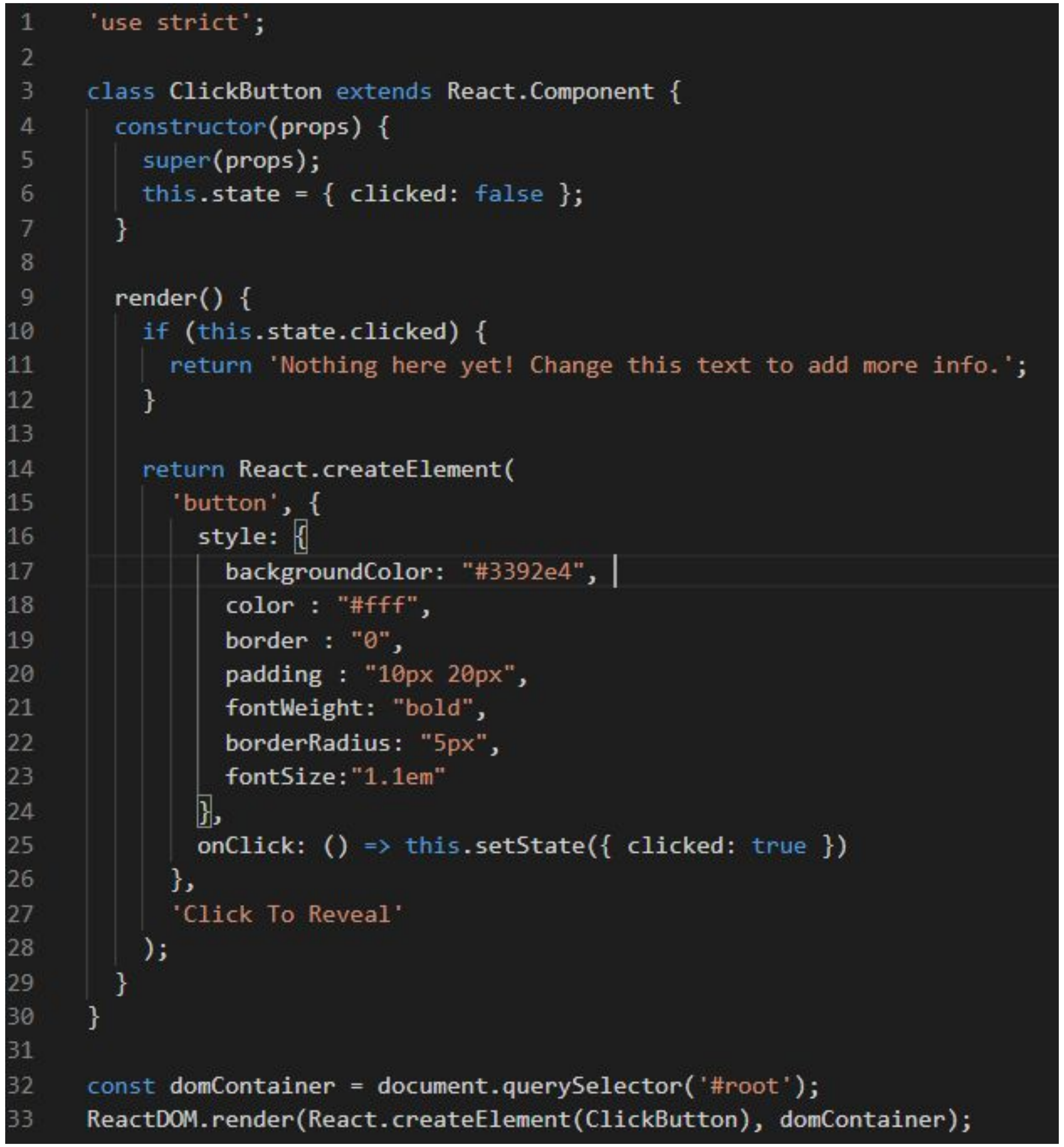
- React components implement a **render()** method that takes input data and returns what to display.



```
'use strict';

class ClickButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { clicked: false };
  }

  render() {
    if (this.state.clicked) {
      return 'Nothing here yet! Change this text to add more info.';
    }

    return React.createElement(
      'button', {
        style: {
          backgroundColor: "#3392e4",
          color : "#fff",
          border : "0",
          padding : "10px 20px",
          fontWeight: "bold",
          borderRadius: "5px",
          fontSize:"1.1em"
        },
        onClick: () => this.setState({ clicked: true })
      },
      'Click To Reveal'
    );
  }
}

const domContainer = document.querySelector('#root');
ReactDOM.render(React.createElement(ClickButton), domContainer);
```

Thinking in React

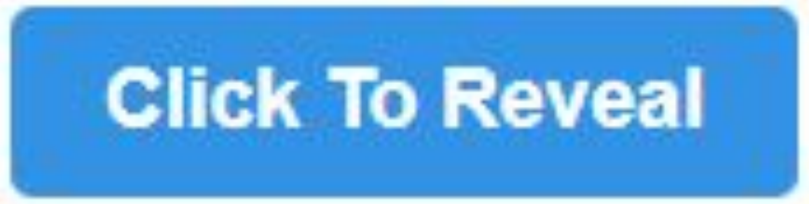
- Line 3 to 30 describe the class.
- Line 14 and 29 returns the React element/component to screen.
- Important to note the component is attached to the **#root** div in the HTML file (Line 32).

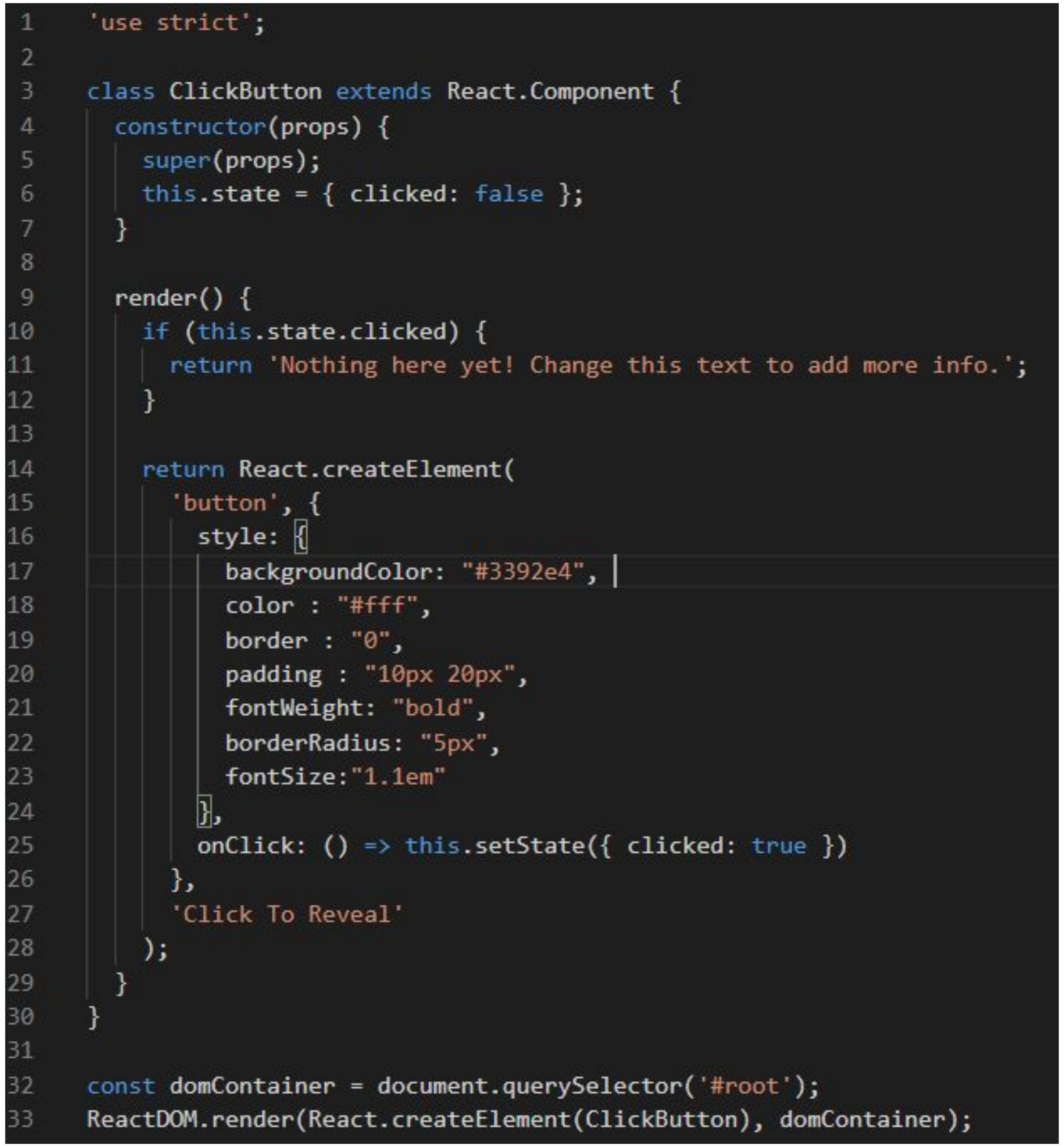


```
1   'use strict';
2
3   class ClickButton extends React.Component {
4     constructor(props) {
5       super(props);
6       this.state = { clicked: false };
7     }
8
9     render() {
10      if (this.state.clicked) {
11        return 'Nothing here yet! Change this text to add more info.';
12      }
13
14      return React.createElement(
15        'button', {
16          style: {
17            backgroundColor: "#3392e4",
18            color : "#fff",
19            border : "0",
20            padding : "10px 20px",
21            fontWeight: "bold",
22            borderRadius: "5px",
23            fontSize:"1.1em"
24          },
25          onClick: () => this.setState({ clicked: true })
26        },
27        'Click To Reveal'
28      );
29    }
30  }
31
32  const domContainer = document.querySelector('#root');
33  ReactDOM.render(React.createElement(ClickButton), domContainer);
```

Thinking in React

- If you view the **index.html** in a browser you will see a button which when clicked will change to the text "*Nothing here yet! Change this text to add more info*"





```
'use strict';

class ClickButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { clicked: false };
  }

  render() {
    if (this.state.clicked) {
      return 'Nothing here yet! Change this text to add more info.';
    }

    return React.createElement(
      'button', {
        style: {
          backgroundColor: "#3392e4",
          color : "#fff",
          border : "0",
          padding : "10px 20px",
          fontWeight: "bold",
          borderRadius: "5px",
          fontSize:"1.1em"
        },
        onClick: () => this.setState({ clicked: true })
      },
      'Click To Reveal'
    );
  }
}

const domContainer = document.querySelector('#root');
ReactDOM.render(React.createElement(ClickButton), domContainer);
```

Checkpoint!

How are you feeling?

RED - I have no idea what you’re talking about.

YELLOW - I have some questions but feel like I understand some things.

GREEN - I feel comfortable with everything you’ve said.

Exercise 1

Exercise 1: Add React to a Web Page

1. Get the latest version of the black-codher-bootcamp repository

2. Open the **index.html** page in **./unit04-react/session1/exercise1** in Visual Code

3. Open the Handout PDF (Session 1) and follow the steps on pages 16 to 22

4. Instructions and code snippets can also be found in the README.md file **./unit04-react/session1/exercise1/README.md**

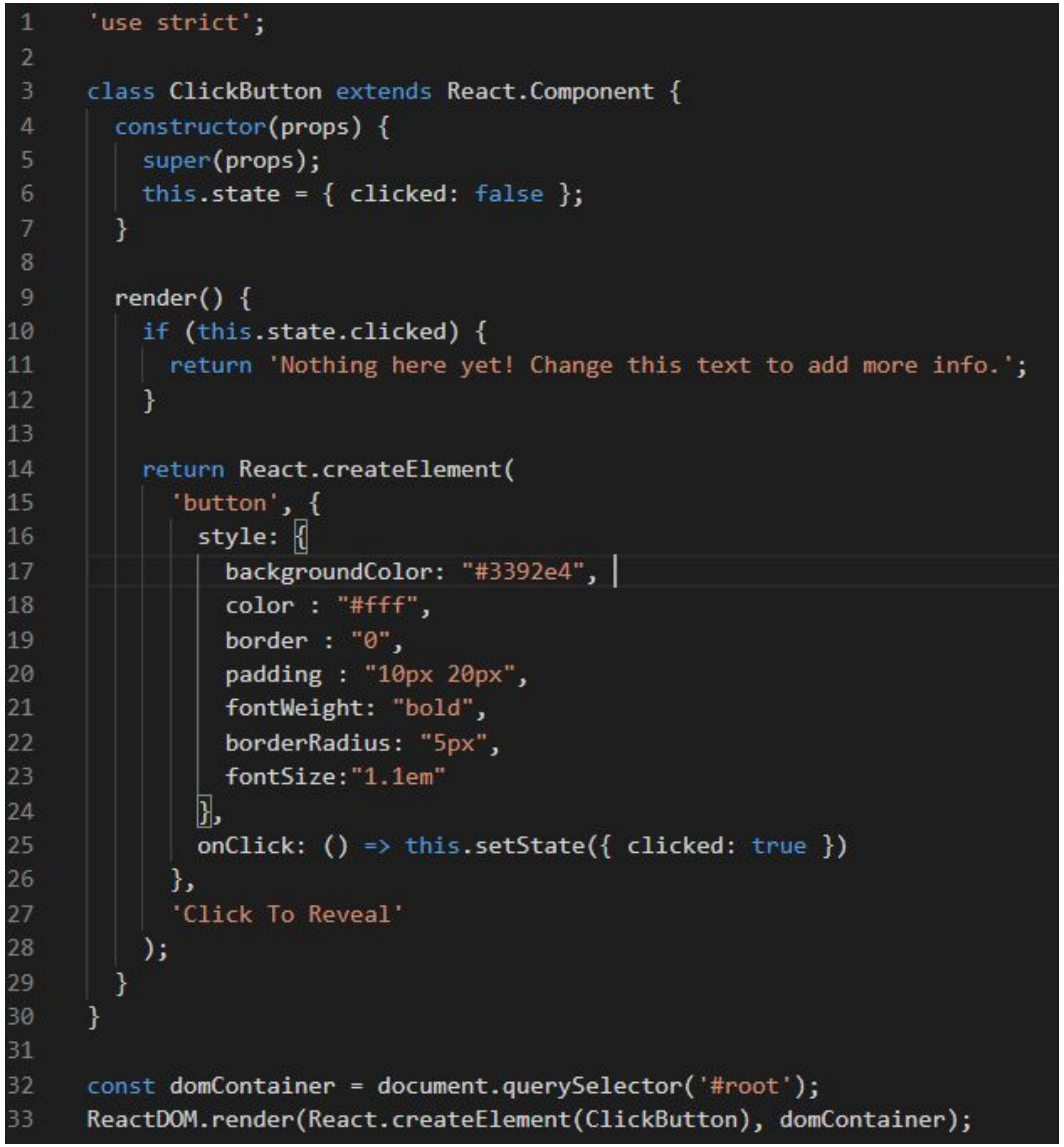
React.createElement Explained

Thinking in React

React.createElement Syntax

```
React.createElement(
  type,
  [props],
  [...children]
)
```

- The type can be a name, string or element.



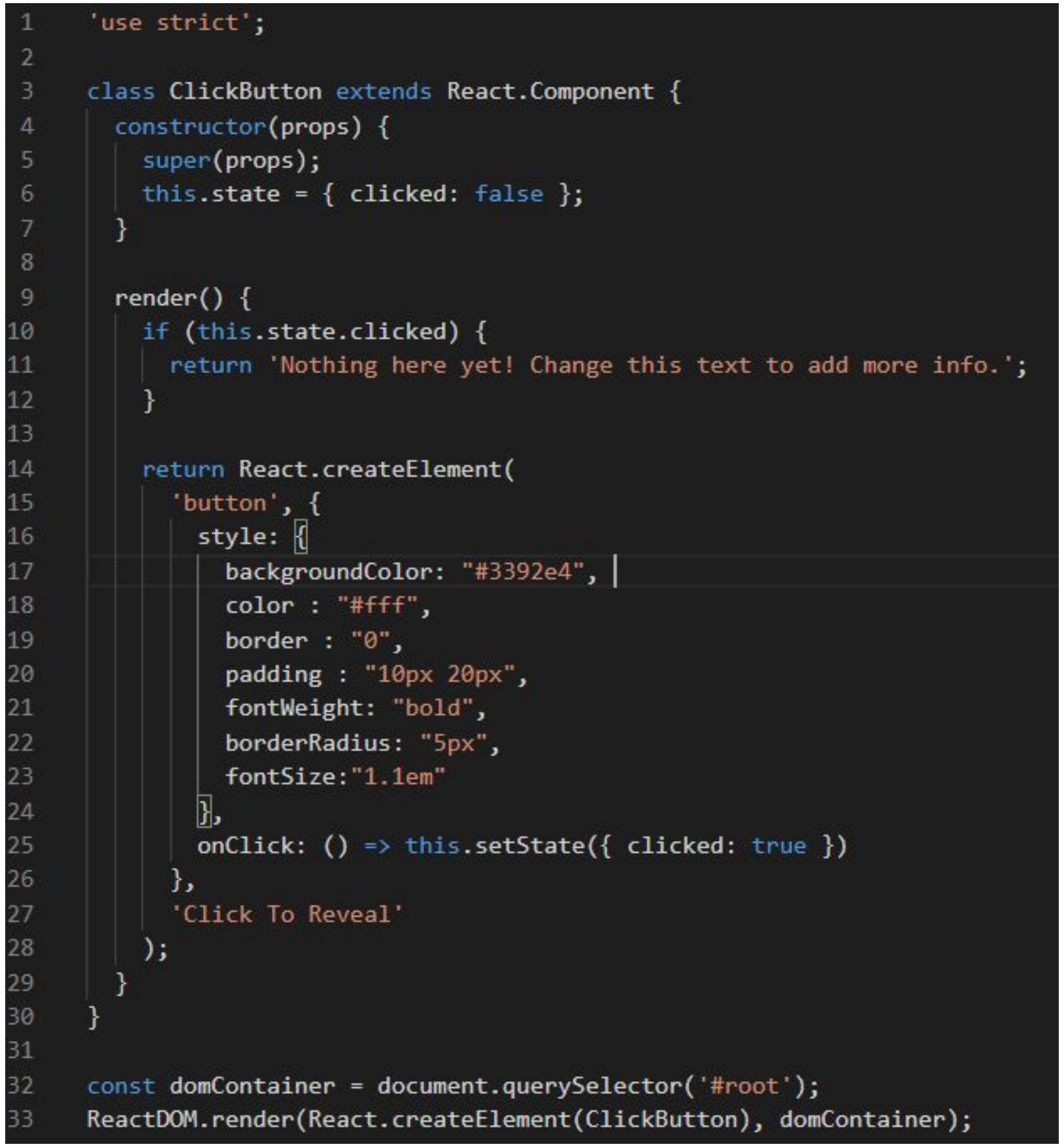
```
'use strict';

class ClickButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { clicked: false };
  }

  render() {
    if (this.state.clicked) {
      return 'Nothing here yet! Change this text to add more info.';
    }

    return React.createElement(
      'button', {
        style: {
          backgroundColor: "#3392e4",
          color : "#fff",
          border : "0",
          padding : "10px 20px",
          fontWeight: "bold",
          borderRadius: "5px",
          fontSize:"1.1em"
        },
        onClick: () => this.setState({ clicked: true })
      },
      'Click To Reveal'
    );
  }
}

const domContainer = document.querySelector('#root');
ReactDOM.render(React.createElement(ClickButton), domContainer);
```

Thinking in React

React.createElement Syntax

```
React.createElement(
   type,
   [props],
   [...children]
)
```

- The **[props]** are attributes of the type such as style, class and events.



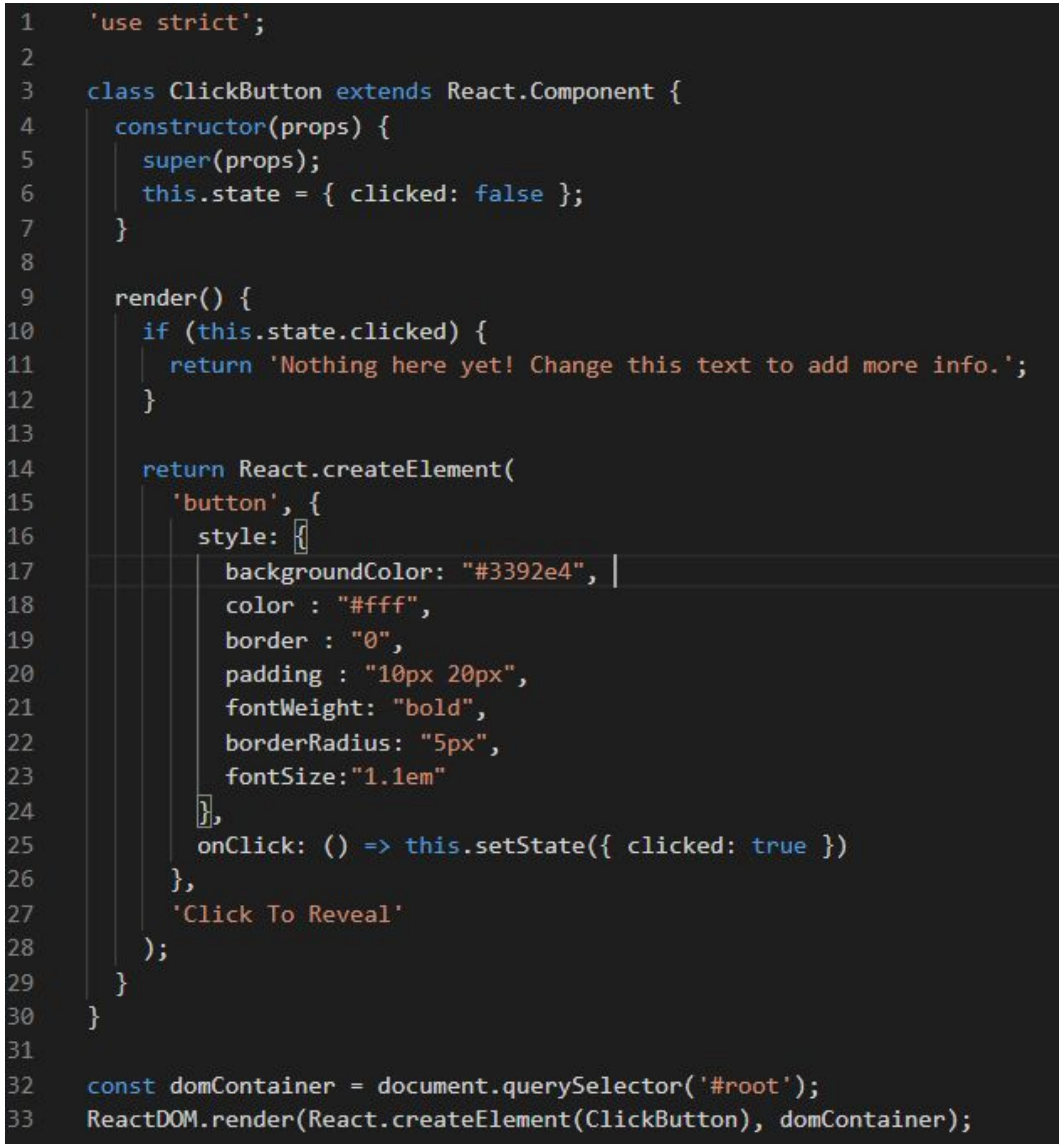
```
'use strict';

class ClickButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { clicked: false };
  }

  render() {
    if (this.state.clicked) {
      return 'Nothing here yet! Change this text to add more info.';
    }

    return React.createElement(
      'button', {
        style: {
          backgroundColor: "#3392e4",
          color : "#fff",
          border : "0",
          padding : "10px 20px",
          fontWeight: "bold",
          borderRadius: "5px",
          fontSize:"1.1em"
        },
        onClick: () => this.setState({ clicked: true })
      },
      'Click To Reveal'
    );
  }
}

const domContainer = document.querySelector('#root');
ReactDOM.render(React.createElement(ClickButton), domContainer);
```

React.createElement

React.createElement Syntax

```
React.createElement(
   type,
   [props],
   [...children]
)
```

- The children are the nested inside the element, in this case the text ‘**Click To Reveal**’ is a child of the button element.



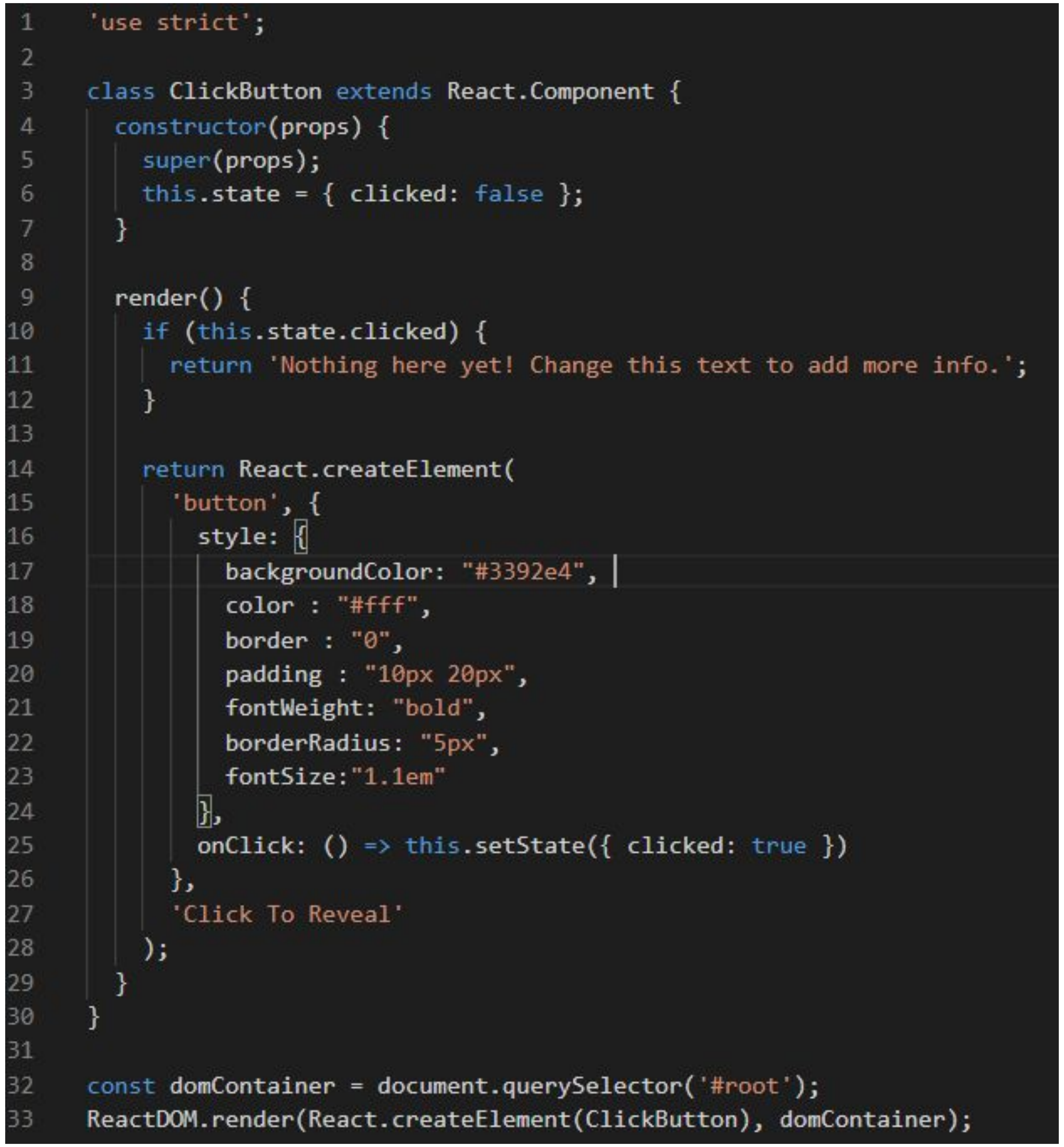
```
'use strict';

class ClickButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { clicked: false };
  }

  render() {
    if (this.state.clicked) {
      return 'Nothing here yet! Change this text to add more info.';
    }

    return React.createElement(
      'button', {
        style: {
          backgroundColor: "#3392e4",
          color : "#fff",
          border : "0",
          padding : "10px 20px",
          fontWeight: "bold",
          borderRadius: "5px",
          fontSize:"1.1em"
        },
        onClick: () => this.setState({ clicked: true })
      },
      'Click To Reveal'
    );
  }
}

const domContainer = document.querySelector('#root');
ReactDOM.render(React.createElement(ClickButton), domContainer);
```

React.createElement

React.createElement Syntax

```
React.createElement(
   type,
   [props],
   [...children]
)
```

- It's important to note that JavaScript styles use **camelCasing** as opposed to **kebab-casing**, so background-color becomes, background**C**olor.



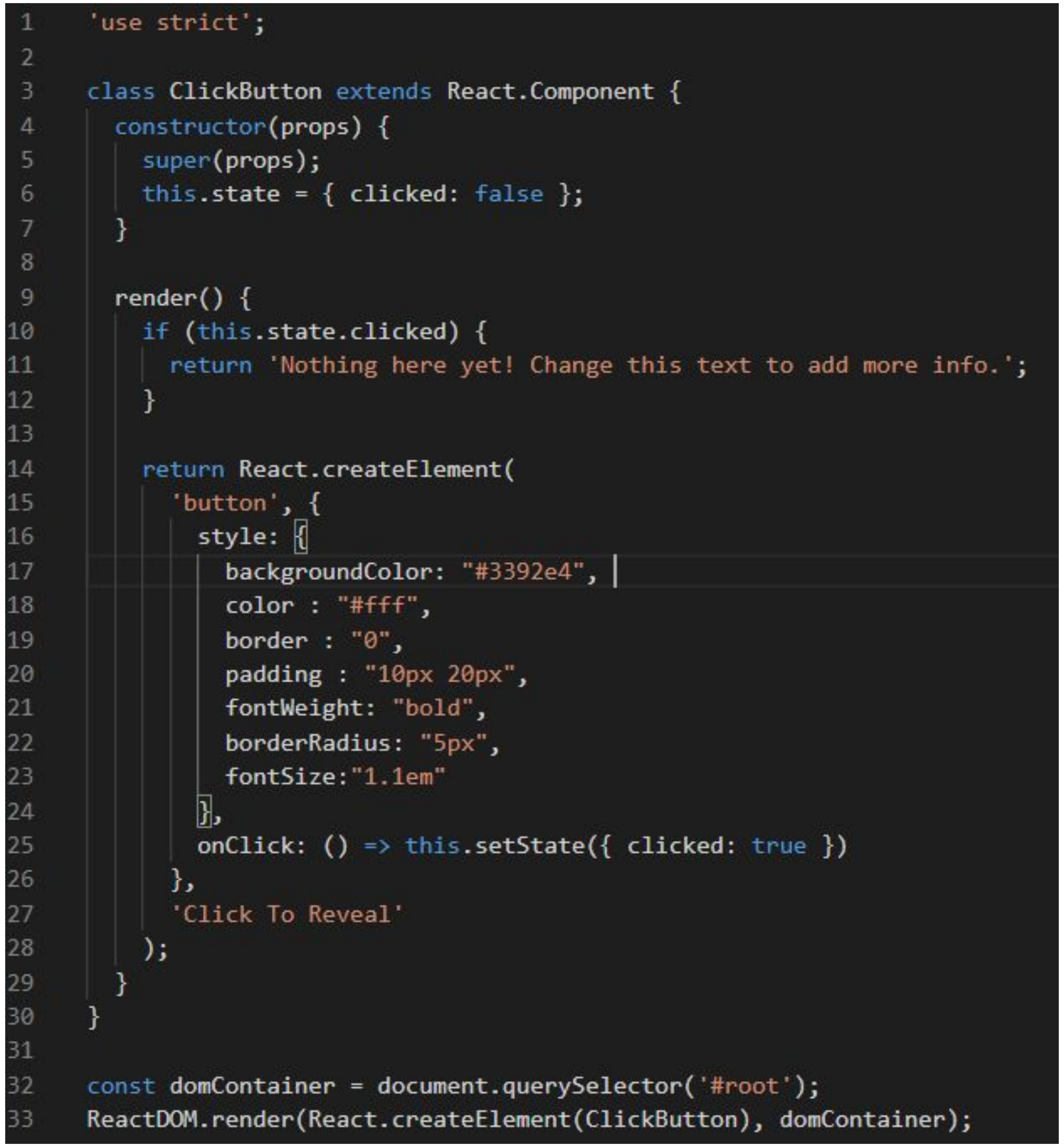
```
'use strict';

class ClickButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { clicked: false };
  }

  render() {
    if (this.state.clicked) {
      return 'Nothing here yet! Change this text to add more info.';
    }

    return React.createElement(
      'button', {
        style: {
          backgroundColor: "#3392e4",
          color : "#fff",
          border : "0",
          padding : "10px 20px",
          fontWeight: "bold",
          borderRadius: "5px",
          fontSize:"1.1em"
        },
        onClick: () => this.setState({ clicked: true })
      },
      'Click To Reveal'
    );
  }
}

const domContainer = document.querySelector('#root');
ReactDOM.render(React.createElement(ClickButton), domContainer);
```

React.createElement

React.createElement Syntax

```
React.createElement(
  type,
  [props],
  [...children]
)
```

- It also important to note that class is a keyword so to add a class attribute to a type/element you will need to use **className**.



```
'use strict';

class ClickButton extends React.Component {
  constructor(props) {
    super(props);
    this.state = { clicked: false };
  }

  render() {
    if (this.state.clicked) {
      return 'Nothing here yet! Change this text to add more info.';
    }

    return React.createElement(
      'button', {
        style: {
          backgroundColor: "#3392e4",
          color : "#fff",
          border : "0",
          padding : "10px 20px",
          fontWeight: "bold",
          borderRadius: "5px",
          fontSize:"1.1em"
        },
        onClick: () => this.setState({ clicked: true })
      },
      'Click To Reveal'
    );
  }
}

const domContainer = document.querySelector('#root');
ReactDOM.render(React.createElement(ClickButton), domContainer);
```

Exercise 2

Exercise 2: Add React to a Web Page

1. Open the **index.html** page in **./unit04-react/session1/exercise2** from the GitHub repository

2. When you open the file in a browser you will see a blue button. Clicking on the button will change the button to text. Change the text in the button (this can be done in the index.js file line 27).

3. Change the text to return a **<h1>** element. Hint: Use a nested **React.createElement**()

Exercise 1: Add React to a Web Page

4. Add a **style** attribute to the returned **<h1>** element, e.g. change it's colour and or size, boldness.

5. Create a new file called **style.css** and move the button css and h1 code into it.

6. Add more data to the page, e.g. small paragraph of text a heading/bio.

Session 1: Summary

- React is a library not a framework. With React you compose user interfaces using small isolated pieces of code called "components”

- React is growing in popularity. Created by Facebook and it is used by multiple large companies

- Learnt the difference between a website, a web application and a progressive web app

- Created a web page referencing the React library

- Explained the **React.createElement()** and it’s components

Checkpoint!

How are you feeling?

RED - I have no idea what you’re talking about.

YELLOW - I have some questions but feel like I understand some things.

GREEN - I feel comfortable with everything you’ve said.